

项目规划书: AI 现金流 (AI Cashflow)

1. 项目概述

- 类型: 策略模拟 / 回合制棋盘游戏
- 核心概念: 结合 **Java Swing** (可视化) 和 **LLM** (无限剧情) 的商业模拟器。
- 一句话介绍: 一个由 AI 担任“上帝”的大富翁游戏, 你的每一个投资机会都是 AI 现场生成的。





2. 核心玩法逻辑 (Gameplay Mechanics)

游戏采用回合制, 核心循环如下:

A. 初始设定

- 玩家开局随机(或自选)一个职业(如:程序员、医生、卡车司机)。
- 每个职业有不同的初始属性:
 - 工资 (Salary): 每经过起点时发放。
 - 初始储蓄 (Cash): 手头的现金。
 - 月支出 (Expenses): 每回合扣除的固定生活费。

B. 回合流程

1. 掷骰子: 玩家点击按钮, 棋子在回字形地图上移动。
2. 触发地块: 根据停留的格子类型, 触发不同逻辑:
 -  **机会格 (Opportunity):**
 - 程序将玩家职业和现金发送给 AI。
 - AI 生成: 一个投资机会(如“隔壁老王急售二手房”或“某科技股大跌”)。
 - 玩家决策: 选择 [买入] 或 [放弃]。
 -  **命运格 (Risk/Doodad):**
 - AI 生成: 一个消费或意外事件(如“突发牙疼需要根管治疗”或“购买新款 iPhone”)。
 - 强制执行: 直接扣除现金, 有时会增加“快乐值”但减少“金钱”。
 -  **市场格 (Market):**
 - AI 发布一条新闻(如“政府宣布降息”)。
 - 这会影响玩家手中已持有资产的价值(这部分如果不做复杂, 可以直接简化为全场资产升值/贬值)。
 -  **银行/发薪日 (Payday):**
 - 经过或停留时, 获得工资, 但需扣除(月支出 + 贷款利息)。

C. 资产管理 (核心策略)

玩家不仅要跑地图, 还要在右侧面板管理自己的资产列表:

- 买入: 消耗现金, 增加“被动收入”(如房租、股息)。
- 卖出: 遇到高价机会时, 卖出资产换取大量现金。

- 贷款: 现金不足时向银行借贷(需每回合付利息)。

3. 胜负判定 (Victory & Defeat)

我们参考《富爸爸穷爸爸》的经典规则, 设定两个阶段的目标:

胜利条件 (如何通关)

目标: 实现财务自由

- 计算公式: 被动收入 (Passive Income) > 总支出 (Total Expenses)
- 解释: 当你不需要工作(工资), 仅靠资产产生的钱(房租、股息)就足够覆盖你的生活开销时, 游戏胜利。
- 进阶胜利(可选): 财务自由后, 购买一个由 AI 生成的“终极梦想”(如“买下一座私人岛屿”)。

失败条件 (Game Over)

结局: 破产

- 触发条件: 现金 (Cash) < 0 且 无法通过变卖资产偿还债务。
- 解释: 当玩家资不抵债, 无法支付每回合的强制支出时, 游戏结束。

4. 技术模块拆解 (分工参考)

A. 界面与交互 (Frontend - Swing)

- 地图绘制: 画出 8x8 或 10x10 的回字形格子, 每个格子有不同的颜色代表类型。
- 动画效果: 棋子移动时要有平滑的过渡动画(使用 Timer), 不能瞬间跳过去。
- 仪表盘: 实时刷新显示: [现金] [被动收入] [总支出] [进度条]。

B. 逻辑与数据 (Backend - Java)

- **Player** 类: 维护 List<Asset> (资产列表) 和 List<Liability> (负债列表)。
- 计算引擎: 每次操作后重新计算 $Cashflow = (工资 + 被动收入) - (支出 + 贷款利息)$ 。

C. AI 大脑 (Intelligence - API)

- **Prompt 工程**: 这是最关键的。我们需要设计好发给 AI 的话术。
 - *Bad Prompt*: "生成一个事件。"
 - *Good Prompt*: "你是一个金融游戏的主持人。玩家职业是医生, 存款5000。请生成一个房地产投资机会, 价格在4000-6000之间, 回报率5%-10%, 并用简短幽默的语言描述。"
- 数据解析: 尝试让 AI 返回 JSON 格式, 或者简单点, 让 AI 在文本最后带上数值 [Cost: 5000, Return: 200], 方便 Java 截取数字进行计算。

5. 总结

这个玩法的核心爽点在于***“未知”***。传统的 Java 练习项目都是死逻辑, 而这个项目因为接入了

AI, 你甚至可以在游戏里跟 AI 讨价还价(如果我们做得快, 可以加上这个功能)。
既有算法逻辑(计算现金流), 又有图形界面(Swing), 还有前沿技术(LLM), 非常适合作为我们的进阶项目。